Efficient Spatio-Temporal Tactile Object Recognition with Randomized Tiling Convolutional Networks in a Hierarchical Fusion Strategy

Lele Cao,[1,2] * **Ramamohanarao Kotagiri,**[2] **Fuchun Sun,**[1] **Hongbo Li,** [1]
Wenbing Huang, [1] and **Zay Maung Maung Aye** [2]
[1] Department of Computer Science and Technology, State Key Lab on Intelligent Technology and Systems,
Tsinghua National Lab for Information Science and Technology (TNList), Tsinghua University, Beijing 100084, China
[2] Department of Computing and Information Systems, The University of Melbourne, Parkville 3010 VIC, Australia

Abstract

Robotic *tactile recognition* aims at identifying target objects or environments from tactile sensory readings. The advancement of unsupervised feature learning and biological tactile sensing inspire us proposing the model of 3T-RTCN that performs *spatio-temporal feature representation and fusion* for tactile recognition. It decomposes tactile data into spatial and temporal threads, and incorporates the strength of *randomized tiling convolutional networks*. Experimental evaluations show that it outperforms some state-of-the-art methods with a large margin regarding *recognition accuracy*, *robustness*, and *fault-tolerance*; we also achieve an order-of-magnitude speedup over equivalent networks with pretraining and fine-tuning. Practical suggestions and hints are summarized in the end for effectively handling the tactile data.

Introduction and Related Work

Recent advances of machine learning techniques and lower-cost sensors have propelled the multidisciplinary research of *robotic recognition*. The *sensory input* plays an essential role in hotspot research areas of *robotic recognition* and *dexterous manipulation*; *tactile sensation* is crucial in particular if visual perception is impaired or non-discriminative. The *tactile sensor* is a device for measuring spatial and temporal property of a physical contact event (Tegin and Wikander 2005). For example (Figure 1), when a robotic hand with tactile sensitivity grasps an object, its tactile sensors output *tactile data* (a.k.a. *tactile sequences*), whose consecutive frames are correlated over time. Tactile data have been explored extensively in object identification (e.g. Madry et al. 2014), stability estimation (e.g. Bekiroglu et al. 2011), slip detection (e.g. Teshigawara et al. 2010), and material awareness (e.g. Chitta, Piccoli, and Sturm 2010). In this paper, we will emphasize *tactile object recognition*, which is to identify target objects from tactile data.

As the foundation of tactile recognition, *spatio-temporal tactile feature representation and fusion* is an active research

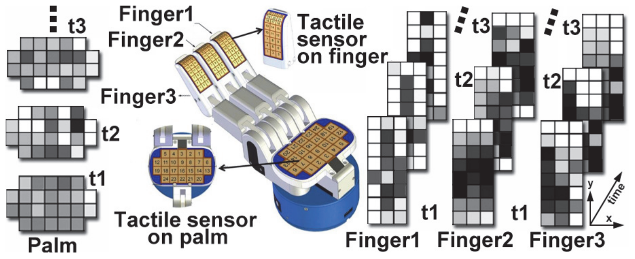


Figure 1: The tactile sequences obtained by a robotic hand.

branch in the domain of *information fusion* (Khaleghi et al. 2013) that has 4 categories (Kokar, Tomasik, and Weyman 2004): *data fusion*, *feature fusion*, *decision fusion*, and *relational information fusion*; the spatio-temporal tactile recognition is actually a practice of the last category, on which researchers have put increasing emphasis recently, attempting to improve recognition ratio and robustness. With a thorough literature survey, we suggest that nearly any spatio-temporal tactile recognition methods could fall into either category of *micro-fusion* or *macro-fusion*. The micro-fusion approaches (Schneider et al. 2009; Taylor et al. 2010; Le et al. 2011; Bekiroglu et al. 2011; Pezzementi, Reyda, and Hager 2011; Liu et al. 2012; Soh, Su, and Demiris 2012; Ji et al. 2013; Drimus et al. 2014; Madry et al. 2014; Xiao et al. 2014; Ma et al. 2014; Yang et al. 2015b) strive to build spatio-temporal joint representation, while macro-fusion approaches (Pezzementi et al. 2011; Simonyan and Zisserman 2014) often construct spatial and temporal features separately, and fuse them in later phases. However, as summarized in (Cao et al. 2015), existing models may suffer from problems such as *task-specific settings*, *variance sensitiveness*, *scale-up limitations*, *manual selection of feature descriptors and distance measures*, and *inferior robustness/fault-tolerance ability*. To address these problems (at least in part), we propose a hybrid architecture that is universal for tactile recognition tasks and has the following advantages over previous work:

- *spatio-temporal decomposition*: considering the nature of tactile data, we invent spatial (*individual frames*) and tem-

*Corresponding author (caoll12@mails.tsinghua.edu.cn).

poral (*tactile flow* and *intensity difference*) threads to describe raw signal efficiently from different perspectives;

- *random feature projection*: to efficiently generate invariant and universal *tactile feature maps*, we build *Randomized Tiling Convolution Network* (RTCN) to convolve and pool each spatial/temporal thread;
- *hierarchical fusion strategy*: for fast recognition, we employ ridge regression for *temporal feature fusion*, extreme learning machine for *spatio-temporal decision fusion*, and 2-layer random majority pooling for *frame-label fusion*.

We formally refer to our approach as "3-Thread RTCN" (3T-RTCN), which has no prior assumption on target-object varieties or hardware settings; hence we intuitively expect it to be more universal than a *task-specific* solution. In RTCN, the pooling and weight-tiling mechanisms allow complex data *invariances* in various scales, while its orthogonal random weights prevent *manual selection of feature descriptors and distance measures*. The non-iterative property of 3T-RTCN, together with RTCN's concepts of local receptive fields and weight-tying, improves the training *efficiency* and *scale-up* capacity significantly. Largely, the strategies of spatio-temporal decomposition and hierarchical fusion may exhibit the *robustness* and *fault-tolerance* ability. The following sections will present RTCN, 3T-RTCN, and our experimental evaluations respectively in detail.

Randomized Tiling ConvNet (RTCN)

RTCN is an extension of ConvNet (Convolutional Network) that consists of two basic operations (convolution and pooling) and two key concepts (local receptive fields and weight-tying). As in Figure 2, the convolution layer may have $F \geq 2$ feature maps to learn highly over-complete representations; the pooling operation is performed over local regions to reduce feature dimensions and hardcode translational invariance to small-scale deformations (Lee et al. 2009). Local receptive fields make each pooling node only sees a small and localized area to ensure computational efficiency and scale-up ability; weight-tying additionally enforces convolutional nodes to share the same weights, so as to reduce the learnable parameters dramatically (Ngiam et al. 2010). RTCN is a ConvNet that adopts mechanisms of *convolutional weights-tiling* and *orthogonalized random weights*.

Convolutional Weights-Tiling and Invariance

The invariance to larger-scale and more global deformations (e.g. scaling and rotation) might be undermined in ConvNet by its constraint to pool over translations of identical invariance. Ngiam et al. (2010) addressed this problem by developing the weights-tiling mechanism (parametrized by a tile size s in Figure 2) which leaves only nodes that are s steps apart to be shared. Weights-tiling is expected to learn a more complex range of invariances because of pooling over convolutional nodes with different basis functions.

In RTCN, we adopt the widely used valid convolution (the weights/filter f is only applied to receptive fields where f fits completely) and square-root pooling. For the t-th tactile frame $\mathbf{x}^{(t)} \in \mathbb{R}^{d \times d}$, each pooling node views a local receptive

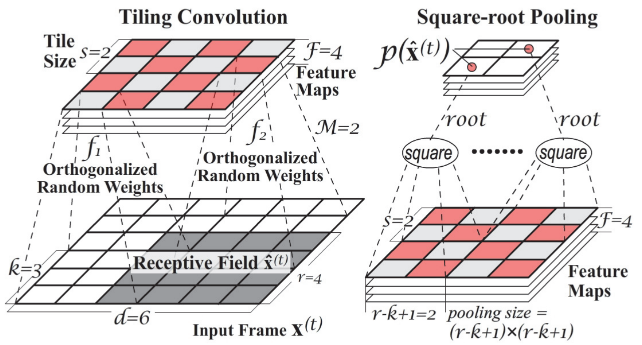

Tiling Convolution
Square-root Pooling
Tile Size
s=2
$\mathcal{F}$=4
Feature Maps
$\mathcal{M}$=2
f_1
f_2
Orthogonalized Random Weights
Orthogonalized Random Weights
k=3
Receptive Field $\hat{\mathbf{x}}^{(t)}$
r=4
d=6
Input Frame $\mathbf{X}^{(t)}$
$p(\hat{\mathbf{x}}^{(t)})$
root
root
square
square
s=2
$\mathcal{F}$=4
Feature Maps
r-k+1=2
pooling size = (r-k+1)×(r-k+1)

Figure 2: A demonstration of RTCN that transforms input frames $(d \times d)$ to feature maps $[(d-r+1) \times (d-r+1)]$.

field $\hat{\mathbf{x}}^{(t)} \in \mathbb{R}^{r \times r}$, $r \leq d$, as shown in Figure 2. With tile size $s \geq 2$, we calculate each pooling node $p(\hat{\mathbf{x}}^{(t)})$ by convolving M different filters $f_1, f_2, \ldots, f_M \in \mathbb{R}^{k \times k}, k \leq r$ with $\hat{\mathbf{x}}^{(t)}$:

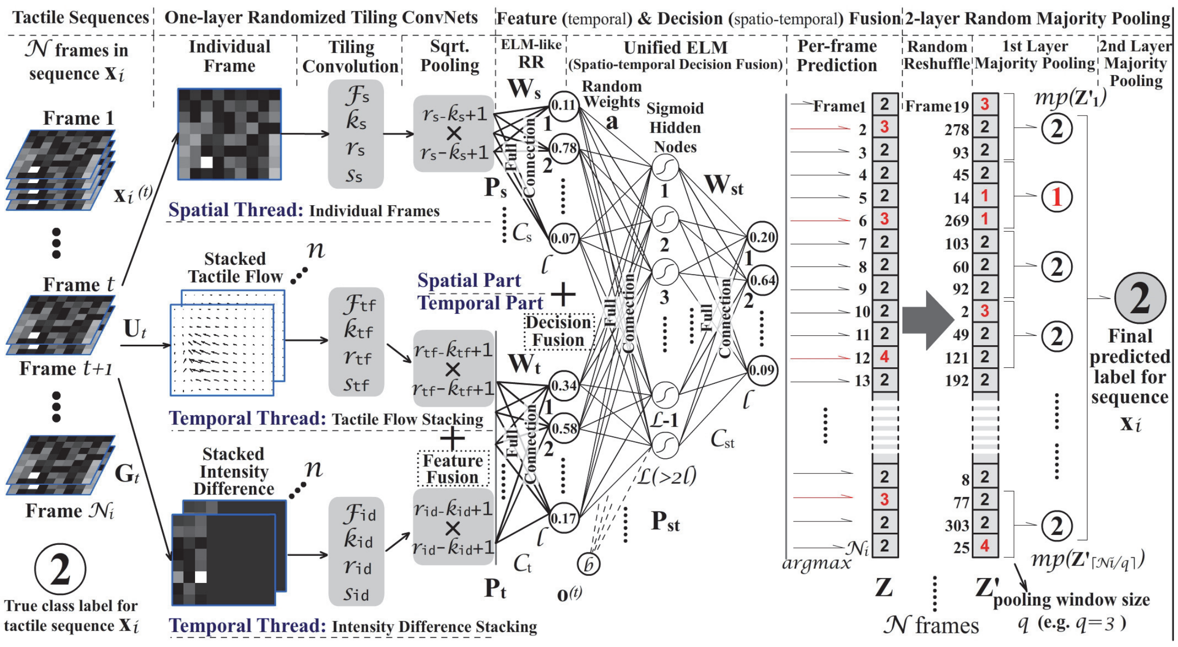
$$p\left(\hat{\mathbf{x}}^{(t)}\right) = \left[\sum_{m=1}^{M} \sum_{i,j \in L_m} \left(f_m *_v \hat{\mathbf{x}}^{(t)}\right)^2_{\langle i,j \rangle}\right]^{\frac{1}{2}}, \quad (1)$$

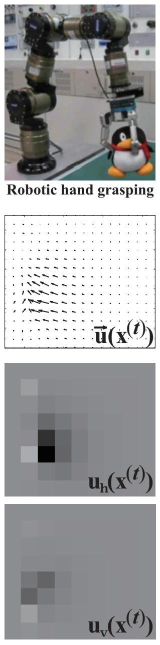
where L_m denotes the set of fields (in form of subscription pairs $\langle i, j \rangle$) where filter f_m should be applied; $*_v$ represents valid convolution operations. By varying s [cf. Figure 5 in (Cao et al. 2015)], we obtain a spectrum of models which trade off between enabling complex invariances and having few filters. A straightforward choice of s would be setting it to the pooling size (i.e. s=r−k+1), so that each pooling node always combines untied convolutional nodes. But a larger s allows more freedom, making it vulnerable to overfitting.

Orthogonalized Random Weights

Tiled ConvNets might require long time to *pretrain and finetune filters* (i.e. *weight-tuning*); this difficulty is further compounded when the network connectivity is extended over the temporal dimension. We noticed several interesting research results (Jarrett et al. 2009; Pinto et al. 2009; Saxe et al. 2011; Huang et al. 2015), which have shown that certain networks with untrained random filters performed almost equally well comparing to the networks with careful weight-tuning. Since any filter f_m in RTCN is shared within the same feature map while distinct among different maps, the initial value of f_m over multiple maps is noted as $\hat{f}_m^{\text{init}} \in \mathbb{R}^{k^2 \times F}, (m=1, \ldots, M)$, which is generated obeying standard Gaussian distribution.

As is suggested in (Ngiam et al. 2010; Huang et al. 2015), $\hat{f}_m^{\text{init}}$ has to be orthogonalized to extract a more complete set of features. However in RTCN, we only need to decorrelate filters (i.e. columns of $\hat{f}_m$) that convolve the same field, because 1) the filters for any two convolutional nodes with non-overlapping receptive fields are naturally orthogonal; and 2) Ngiam et al. (2010) empirically discovered that "orthogonalizing partially overlapping receptive fields is not necessary for learning distinct, informative features". Hence this local orthogonalization is computationally cheap using SVD: the columns of $\hat{f}_m$ are the orthonormal basis of $\hat{f}_m^{\text{init}}$.


Tactile Sequences
One-layer Randomized Tiling ConvNets
Feature (temporal) & Decision (spatio-temporal) Fusion
2-layer Random Majority Pooling
Spatial Thread: Individual Frames
Temporal Thread: Tactile Flow Stacking
Temporal Thread: Intensity Difference Stacking
Unified ELM (Spatio-temporal Decision Fusion)
Per-frame Prediction
Random Reshuffle
1st Layer Majority Pooling
2nd Layer Majority Pooling
Final predicted label for sequence $\mathbf{x}_i$
True class label for tactile sequence $\mathbf{x}_i$
pooling window size q (e.g. $q=3$)


Robotic hand grasping
$\vec{\mathbf{u}}(\mathbf{x}^{(t)})$
$\mathbf{u}_h(\mathbf{x}^{(t)})$
$\mathbf{u}_v(\mathbf{x}^{(t)})$

(a) 3T-RTCN: we only use $\mathbf{x}_i$ for explanation; but it does not distinguish different sequences until majority pooling. (b) Tactile flow

Figure 3: The illustration of (a) 3T-RTCN architecture and (b) tactile flow between two neighbouring frames $\mathbf{x}^{(t)}$ and $\mathbf{x}^{(t+1)}$.

The Proposed Architecture: 3T-RTCN

3T-RTCN is built on the decomposition of tactile sequences into spatial and temporal parts. Probably owing to **relatively lower dimensionality and less diversity of tactile frames than videos**, we empirically found that using multiple layers of RTCN basically adds no positive contribution to recognition ratio; thus as illustrated in Figure 3a, one-layer RTCN is employed in each thread. With the same notations defined previously, the spatial thread is parameterized by a quadruplet (F_s, k_s, r_s, s_s); and two temporal threads are parameterized by $(F_{tf}, k_{tf}, r_{tf}, s_{tf})$ and $(F_{id}, k_{id}, r_{id}, s_{id})$. In the upcoming sections, tactile frames for training are formalized as $\aleph = \{(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}) | \mathbf{x}^{(t)} \in \mathbb{R}^{d\times d}, \mathbf{y}^{(t)} \in \mathbb{R}^l, t = 1, \ldots, N\}$.

Spatial Thread: Individual Frames

Based on the belief that the static tactile force distribution is an informative cue by itself, the *spatial thread* is designed to operate on individual tactile frames. The output of spatial RTCN forms a spatial feature space $\mathbf{P}_s$:

$$\mathbf{P}_s = \begin{bmatrix} \mathbf{p}(\mathbf{x}^{(1)}) \\ \vdots \\ \mathbf{p}(\mathbf{x}^{(N)}) \end{bmatrix} = \begin{bmatrix} p_1(\mathbf{x}^{(1)}) & \cdots & p_F(\mathbf{x}^{(1)}) \\ \vdots & \vdots & \vdots \\ p_1(\mathbf{x}^{(N)}) & \cdots & p_F(\mathbf{x}^{(N)}) \end{bmatrix}. \quad (2)$$

The frame $\mathbf{x}^{(t)}$ is fed to an RTCN with F maps to generate a *joint pooling activation* $\mathbf{p}(\mathbf{x}^{(t)}) = [p_1(\mathbf{x}^{(t)}), \ldots, p_F(\mathbf{x}^{(t)})]$, which is a row vector concatenating *pooling outputs*. $p_i(\mathbf{x}^{(t)})$ is also a row vector with $(d–r+1)^2$ pooling activations for the i-th feature map. The row vectors of $\mathbf{P}_s$ are in full connection (weighted by $\mathbf{W}_s$ in Figure 3a) with l output nodes representing l object classes. Principally, $\mathbf{W}_s \in \mathbb{R}^{[F\times(d-r+1)^2]\times l} = [\mathbf{w}_1, \ldots, \mathbf{w}_{F\times(d-r+1)^2}]^T$ can be learned with any supervised learning algorithm such as Ridge Regression (RR). We follow (Huang et al. 2012) to tune $\mathbf{W}_s$ because of its fast training speed and good generalization ability:

$$\mathbf{W}_s = \begin{cases} \mathbf{P}_s^T \left(\frac{\mathbf{I}}{C} + \mathbf{P}_s\mathbf{P}_s{}^T\right)^{-1}\mathbf{Y}, & N \le F\times(d-r+1)^2 \\ \left(\frac{\mathbf{I}}{C} + \mathbf{P}_s^T\mathbf{P}_s\right)^{-1}\mathbf{P}_s^T\mathbf{Y}, & N > F\times(d-r+1)^2 \end{cases}, \quad (3)$$

where $\mathbf{Y}$ is defined as $[\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(N)}]^T_{N\times l}$; small positive values $\mathbf{I}/C$ is added to improve result stability; when the training set is very large ($N \gg$ size of feature space), the second solution should be applied to reduce computational costs.

Temporal Thread: Tactile Flow Stacking

The term of *tactile flow* was initially introduced by Bicchi et al. (2008) for analyzing tactile illusory phenomena; it is intimately related to the vision models of optical flow, inspired by which, we calculate the inter-frame tactile flow, and use it as a temporal feature descriptor. Since tactile readings can be **hardly affected by factors like color and illumination**, we empirically discover that some complex optical flow models (e.g. Sun, Roth, and Black 2010) might have negative impact for tactile recognition and demand more computation effort. So we simply follow (Horn and Schunck 1981), and perform *mean flow subtraction* and *tactile flow stacking*.

As shown in Figure 3b, we use $\vec{\mathbf{u}}(\mathbf{x}^{(t)})$ to denote the two-dimensional vector field of tactile flow between frames $\mathbf{x}^{(t)}$ and $\mathbf{x}^{(t+1)}$; and use notation $\vec{\mathbf{u}}(x^{(t)}_{i,j})$ to indicate the vector at point $\langle i,j\rangle_{i,j=1,\ldots,d}$. The horizontal and vertical components, $\mathbf{u}_h(\mathbf{x}^{(t)})$ and $\mathbf{u}_v(\mathbf{x}^{(t)})$, are treated as two channels in the temporal RTCN. The overall tactile flow can be dominated by a particular direction (e.g. caused by sudden slippage in an unstable grasping), which is unfavoured to ConvNets. As zerocentering the input of ConvNets allows better exploiting the rectification nonlinearities (Simonyan and Zisserman 2014), we perform mean flow subtraction by subtracting the mean vector from $\bar{\mathbf{u}}(\mathbf{x}^{(t)})$. To represent force motion across several consecutive frames, we stack n vector fields together; hence the stacked tactile flow $\mathbf{U}_t$ for frame $\mathbf{x}^{(t)}$ is formulated as

$$\mathbf{U}_t(i,j,2t'{-}1) = u_{\mathrm{h}}\Big(x_{i,j}^{(t+t'-1)}\Big), \mathbf{U}_t(i,j,2t') = u_{\mathrm{v}}\Big(x_{i,j}^{(t+t'-1)}\Big), \quad (4)$$

where $\mathbf{U}_t(i,j,2t')_{t'=1,\dots,n}$ denotes the force motion at point $\langle i,j\rangle$ over a sequence of n frames; $\mathbf{U}_t$ has $2n$ input channels.

Temporal Thread: Intensity Difference Stacking

Intensity difference has been widely used to detect moving targets in video clips by carrying out differential operation on every two successive frames, which may do well in ideal condition, but it is especially susceptible to the variation of illumination light. **Unlike video clips, tactile readings only respond to grip force**. So tactile intensity difference is also an informative temporal cue to describe the change of force intensity and distribution. To remove the noise residual, we perform the high-pass filtering with a threshold T. $\mathbf{g}(\mathbf{x}^{(t)})$ is used to denote the tactile difference between $\mathbf{x}^{(t)}$ and $\mathbf{x}^{(t+1)}$; and the value change of unit $\langle i,j\rangle$ is set to zero unless

$$g(x_{i,j}^{(t)}) = x_{i,j}^{(t)} - x_{i,j}^{(t+1)}, \left|x_{i,j}^{(t)} - x_{i,j}^{(t+1)}\right| \geq T. \quad (5)$$

We can also stack n consecutive intensity differences to represent changes over a larger granularity; the stacked volume for frame t is denoted as $\mathbf{G}_t$ that embodies n channels:

$$\mathbf{G}_t(i,j,t') = g\left(x_{i,j}^{(t+t'-1)}\right), \; t' = 1,\dots,n. \quad (6)$$

Hierarchical Fusion Strategy

Temporal Feature Fusion The output vectors from pooling layers in both temporal threads are concatenated constituting a *joint temporal feature space* $\mathbf{P}_{\mathrm{t}}$, and the row vectors of which are fully connected (weighted by $\mathbf{W}_{\mathrm{t}}$) to l output nodes. $\mathbf{W}_{\mathrm{t}}$ is initialized randomly and tuned with Eq. (3).

Spatio-Temporal Decision Fusion The l output probability values from spatial and temporal part are concatenated to compose a *joint spatio-temporal decision space* with $2l$ dimensions, which is noted as $\mathbf{o}^{(t)} \in \mathbb{R}^{2l}$ in Figure 3a. To learn a complex nonlinear function to fuse the spatial and temporal decisions, an Extreme Learning Machine (ELM) with L hidden nodes and l output nodes is applied. We simply use the sigmoidal activation due to its proved universal approximation and classification capability (Huang et al. 2015); the activation function of the i-th hidden node is

$$Sig(\mathbf{a}_i, b_i, \mathbf{o}^{(t)}) = \left[1 + e^{-\left(\mathbf{a}_i \cdot \mathbf{o}^{(t)} + b_i\right)}\right]^{-1}, \quad (7)$$

where the parameters $\{\mathbf{a}_i, b_i\}_{i=1\cdots L}$ are randomly generated obeying any continuous probability distribution (Huang et al. 2012). The spatio-temporal feature space is defined as

$$\mathbf{P}_{\mathrm{st}} = \begin{bmatrix} Sig(\mathbf{a}_1, b_1, \mathbf{o}^{(1)}) & \cdots & Sig(\mathbf{a}_L, b_L, \mathbf{o}^{(1)}) \\ \vdots & \ddots & \vdots \\ Sig(\mathbf{a}_1, b_1, \mathbf{o}^{(N)}) & \cdots & Sig(\mathbf{a}_L, b_L, \mathbf{o}^{(N)}) \end{bmatrix}. \quad (8)$$

Optimal L is selected from $\{100, 200, 300, 400, 500\}$; and spatio-temporal weight matrix $\mathbf{W}_{\mathrm{st}}$ is adjusted with Eq. (3).

Table 1: Tactile datasets. **SDH: Schunk Dexterous Hand, SPG: Schunk Parallel Gripper, BDH: Barrett Dexterous Hand**.

Data Specification	SD10	SPr10	BDH10	HCs10
Frame Dimension	13×18	8×16	8×13	4×4
# Object Classes	10	10	10	10
# Tactile Seq.	100	97	53	180
Mean Seq. Length	349	511	561	37
# Training Frames	25,001	36,782	24,253	4,720
# Testing Frames	9,857	12,787	5,465	1,812

Frame-Label Fusion: 2-Layer Random Majority Pooling Defining $\mathrm{Sig}(\mathbf{a}, \mathbf{b}, \mathbf{o}^{(t)})$ as the row vector of $\mathbf{P}_{\mathrm{st}}$, we predict the class label for the t-th frame $\mathbf{x}^{(t)}$ using

$$label(\mathbf{x}^{(t)}) = \arg \max_{y \in \{1,\dots,l\}} \Big[\mathrm{Sig}(\mathbf{a}, \mathbf{b}, \mathbf{o}^{(t)}) \cdot \mathbf{W}_{\mathrm{st}}\Big]. \quad (9)$$

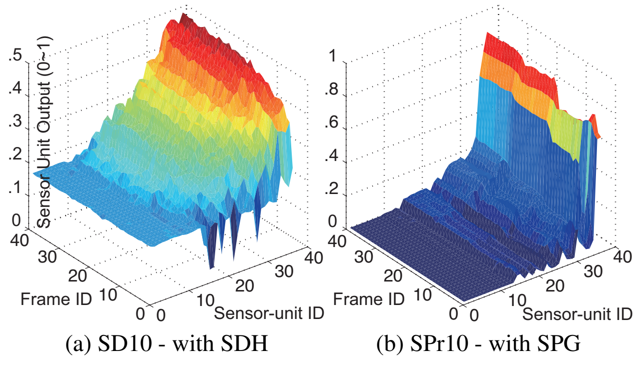
In this way, we generate N_i labels for the i-th sequence $\mathbf{x}_i$ that has N_i frames. Considering that frames in the same tactile sequence are obtained via manipulating the same object, we use 2-layer (applying more than one layer enables certain prediction invariance and stability) random majority pooling to predict sequence labels. Specifically, the N_i frame labels (denoted as a set $\mathbf{Z}$) are randomly reshuffled, yielding a new set $\mathbf{Z}'$. We then in the first layer use a majority pooling window of size q sliding in a non-overlapping manner, in which it votes for the most-frequent label. The first layer generates $\lceil \frac{N_i}{q} \rceil$ labels, which are majority-pooled in the second layer: $label(\mathbf{x}_i) = mp[mp(\mathbf{Z}'_1), \dots, mp(\mathbf{Z}'_{\lceil N_i/q \rceil})]$, where $mp(\bullet)$ denotes the operation of majority pooling.

Experiments with Real-World Tasks

In this section, we evaluate 3T-RTCN on several benchmark datasets: SD5, SD10 (Bekiroglu, Kragic, and Kyrki 2010; Bekiroglu et al. 2011), SPr7, SPr10 (Drimus et al. 2014), BDH5 (Yang et al. 2015b), BDH10 (Xiao et al. 2014), and HCs10 (Yang et al. 2015a), which represent different tactile recognition tasks with various complexities. SD5, SPr7, and BDH5 turn out to be similar to SD10, SPr10, and BDH10 respectively in many ways, so here we only focus on 4 datasets specified in Table 1. The parameters (i.e. F, k, r, s, C, L) were determined by a combination of grid search and manual search (Hinton 2012) on validation sets. The parallel architecture of 3T-RTCN, shallow structure of RTCNs (without weight-tuning), and efficient fusion strategies make the parameter evaluation extremely fast.

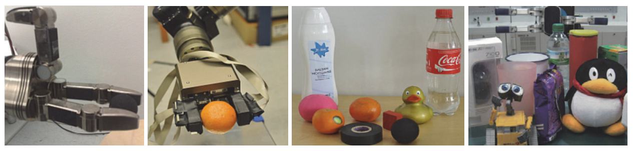
Briefs of Tactile Benchmark Datasets

The SD5 and SPr10 datasets were collected with the 3-finger SDH (Figure 5a) and the 2-finger SPG (Figure 5b) respectively; the grasp execution applied was similar: a household object (cf. Figure 5c) was manually placed between the fingers at the beginning; after the first physical contact with the object, the fingers moved back and forth slightly for 5 times, and then released the object in the end. The BDH10 dataset was collected with a 4-degree-of-freedom (4-DoF) BDH mounted on the end of a 7-DoF Schunk manipulator (the uppermost image in Figure 3b); the hand (cf. Fig-



(a) SD10 - with SDH (b) SPr10 - with SPG

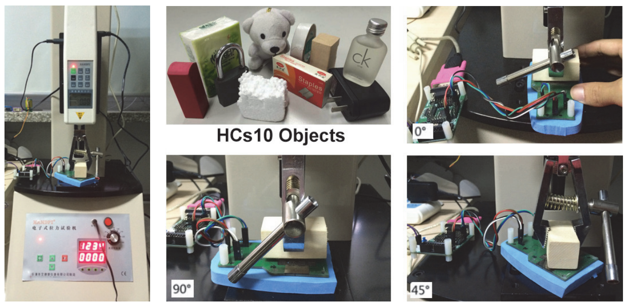
Figure 4: The visualization of a tactile sequence obtained by grasping the same object using different grippers: the units are sorted by their mean output over the temporal axis.



(a) SDH (b) SPG (c) SD/SPr10 obj. (d) BDH10 obj.

Figure 5: Robotic hands and/or objects used to collect SD10 (a+c), SPr10 (b+c), and BDH10 datasets: (a) one 13×6 sensor per finger; (b) one 8×8 flexible sensor per finger.

ure 1) has 3 fingers with one sensor patch (3×8) mounted on each finger and one (24 units) on the palm; ten objects (Figure 5d) with different shape, hardness, and surface texture are grasped (from the initial contact till the complete cease of the lifting movement) for 5∼6 times each, bringing about 53 tactile sequences in total. The HCs10 dataset was collected from a flexible capacitive tactile sensor array on the Handpi electronic test stand (Figure 6a); one of the ten objects (top-left of Figure 6b) was fastened at the tail of the force gauge with a clamp and moved down towards the sensor with a small fixed speed; the data was logged from the initial contact until the time when normal force measured by the dynamometer reached 5 kg·m/s^2; each object was placed in 3 directions as shown in Figure 6b, repeating 6 times (i.e. 3 stable contacts and 3 contacts with slippage) for each direction, so as to get 18 sequences for each object.



(a) Handpi-50 (b) Objects and contact/sliding directions

Figure 6: The illustration of collecting the HCs10 dataset.

On Spatial and Temporal Threads

The intention of this section is to validate the performance of 3T-RTCN compared to that of using either spatial or temporal information only, and also to clarify the contributions of spatial and temporal information to the overall performance. The accuracies in Figure 7 were averaged over 10 trials, each of which used a 7:3 train/test-ing split. 3T-RTCN achieved 100% recognition ratio for all datasets except HCs10, which might be yet another circumstantial evidence of (Anselmi et al. 2015) that invariant features cut the requirement on sample complexity; it phenomenally matches people's ability to learn from very few labeled examples. To locate the factors that influence the discrimination ability of spatial and temporal information, we visualize the output of maximally activated sensor units when grasping the same object using SDH (Fig. 4a) and SPG (Fig. 4b); we discovered that SPG produced sparse (spatial) and smooth (temporal) signal resembling certain filtering effect, while SDH generated dense and spiky output. We believe that temporal threads tend to perform better for sparse and filtered tactile data, while spatial ones becomes relatively more crucial when the task involves more contact areas (i.e. more tactile sensor units with dense output). It is noteworthy that increasing the stacking size n from 2 to 5 only leads to a trivial enhancement or even worse performance, so we use $n = 2$ for the experiments followed.

On Weight-Tuning and Micro-Fusion

The unexpectedly high recognition ratio of 3T-RTCN makes us curious about the role of *weight-tuning*. So we pretrained (Ngiam et al. 2010) and finetuned (Schmidt 2012) the convolutional weights of 3T-RTCN. Fig. 8a illustrates that weight-tuning invariably increased the accuracy of frame-wise prediction by an average margin of 1.01%; but this gain was not big enough to differentiate the sequence-wise performance; thus a properly parameterized architecture is as beneficial as weight-tuning. Observed from Figure 8b, the weight-tuning stepped up the averaged training time by roughly 25 times, which is unfavoured in real-time tactile recognition tasks.

While 3T-RTCN falls into the macro-fusion category, we are also interested in checking out the performance of micro-fusion ConvNets that capture the motion information at an early stage. In this respect, we tested the so-called 3D ConvNet (Ji et al. 2013) which convolves a 3D filter with a cube formed by 7 contiguous frames; each cube has 25 maps in 4 channels: 7 (Frames) + 12 (TactileFlows: 6 horizontal and 6 vertical) + 6 (IntensityDifferences). The convolutional weights and output weights were trained by (Schmidt 2012). As can be seen from Fig. 8a, 3T-RTCN consistently outperformed the finetuned (no pretraining) 3D ConvNet with an average gap of 5.2%, underpinning the effectiveness of higher-level feature/decision fusion. Notably in Fig. 8b, 3T-RTCN acted 6∼10 times faster than the 3D ConvNet, which is partly due to the reduced output dimensionality of square-root pooling.

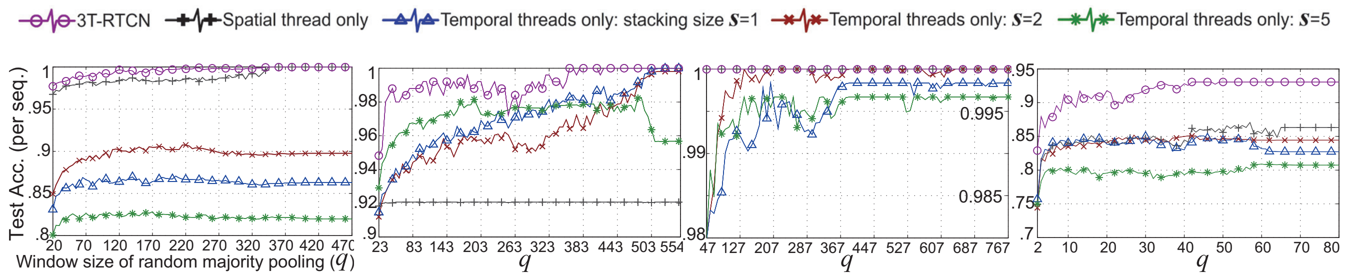
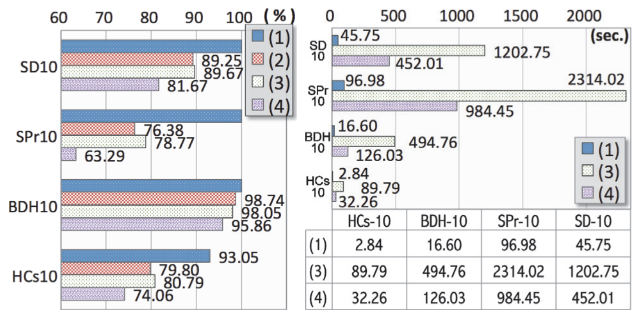


Figure 7: Mean accuracy as a function of window size for random majority pooling. Left to right: SD10, SPr10, BDH10, HCs10.

Table 2: The comparison of average recognition ratio between 3T-RTCN and the state-of-the-arts. (direct-quote*, **best**, <u>2nd best</u>)

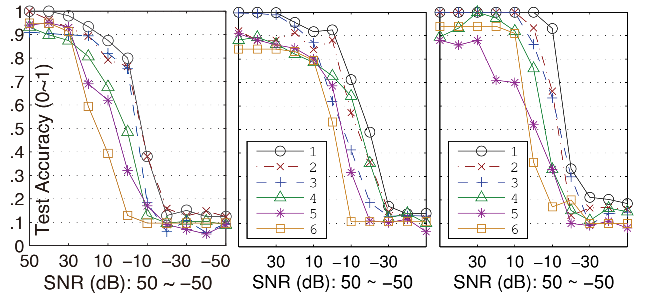
Datasets/Models	3T-RTCN	JKSC	MV-HMP	ST-HMP	LDS-Martin	BoS-LDS	Other Models
SD10	**100.0**	91.5	94.0 *	94.0 *	92.0	<u>97.5</u> *	97.0 * (MV-HMP$_{\rm FD}$)
SPr10	**100.0**	87.0 *	84.5 *	88.5 *	<u>94.5</u>	94.2	91.1 * (ST-HMP$_{\rm FD}$)
BDH10	**100.0**	<u>94.0</u>	81.6	87.5	82.0	90.5	96.0 * (pLDSs)
HCs10	<u>91.2</u>	**93.5**	67.7	83.0	70.0	74.5	N/A



	HCs-10	BDH-10	SPr-10	SD-10
(1)	2.84	16.60	96.98	45.75
(3)	89.79	494.76	2314.02	1202.75
(4)	32.26	126.03	984.45	452.01

(a) Avg. test accuracy (%) (b) Avg. training time (sec.)

Figure 8: Comparison of 4 approaches: (1) 3T-RTCN, (2) the 3T-RTCN without majority pooling, (3) same as "(2)" with pretrain & finetune, and (4) micro-fusion with 3D ConvNets.



(a) SD10 (b) SPr10 (c) BDH10

Figure 9: Noise robustness: 1) 3T-RTCN, 2) spatial thread, 3) temporal threads, 4) ST-HMP, 5) LDS-Martin, 6) JKSC.

On Comparison with State-of-the-Arts

Here we compare 3T-RTCN with state-of-the-art models for tactile recognition; our study involves both micro and macro fusion methods, which include LDS-Martin (Saisan et al. 2001), ST/MV-HMP, ST/MV-HMP$_{\rm FD}$ (Madry et al. 2014), BoS-LDSs (Ma et al. 2014), pLDSs (Xiao et al. 2014), and JKSC (Yang et al. 2015b). For fair comparability with some directly referenced results (i.e. the asterisked scores in Table 2), all simulations in this section were carried out on 10-fold cross validations with 9:1 splits. The highest score for each dataset is indicated in bold; and the second best one is underlined. The 3T-RTCN achieved the best recognition rate on all tasks except HCs10. For low-dimensional tactile data (e.g. HCs10), convolution and pooling will cause a big granularity loss, which may undermine the recognition ratio slightly; nevertheless, we still have reasons (e.g. efficiency, robustness, and fault-tolerance) to apply 3T-RTCN on low-dimension dataset. Additionally, we noticed empirically that JKSC and MV/ST-HMP were constantly trained hundreds of times slower than our model, which was majorly induced by the time-consuming activities of kernel computation and dictionary learning respectively.

On Robustness and Fault-Tolerance

Robustness and fault-tolerance both describe the consistency of systems' behavior, but *robustness* describes the response to the input, while *fault-tolerance* describes the response to the dependent environment. Johnson (1984) defined a fault-tolerant system as the one that can continue its intended operation (possibly at a reduced level) rather than failing completely, if the system partially fails. We will compare with 3 models: JKSC, ST-HMP, and LDS-Martin, which represent 3 mainstream methodologies: sparse coding, unsupervised feature learning, and time-series modeling.

Robustness to Sensor Noise We manually added different noise capacities (white Gaussian) to frame vectors. SNR (Signal-to-Noise Ratio) is used as a measure for comparing the strength of the desired force signal to the level of background noise. It is defined as the ratio of signal strength to

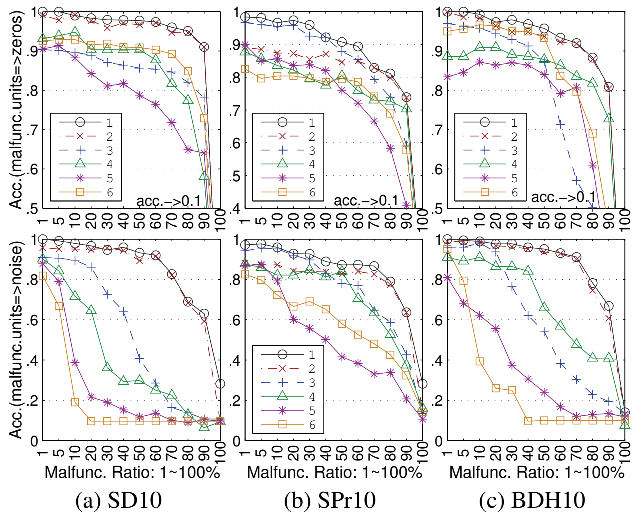


Figure 10: Fault-tolerance to sensor malfunc. Legend: Fig. 9.

noise power, and often expressed in decibels (dB): $\text{SNR}_{\text{dB}} = 10 \cdot \log_{10}(P_{\text{sig}}/P_{\text{nos}})$, where P_{sig} and P_{nos} denote the power of signal and noise respectively; and SNR>0 indicates more signal than noise. Figure 9 demonstrates that our way of fusing the strength of spatial and temporal information offered the best noise robustness quality. 3T-RTCN still maintained strong classification capability even when the noise strength ratio reached up to 50% (SNR=0); but with the same amount of noise contamination, other models' performance dropped dramatically without exception. ST-HMP turned out to have the best noise robustness among the three referenced methods, while JKSC had relatively the worst one that sometimes lost its classification power completely at SNR=0.

Fault-Tolerance to Partial Sensor Malfunction A common hardware glitch of tactile sensors is *partial sensor malfunction*, in which one or more units composing the sensor array are broken and hence always output a fixed value (most likely to be zero) or random values; and it is prone to occur in extreme environments. To examine the fault-tolerance ability in coping with such failure, we intentionally sabotaged a percentage ($1 \sim 100\%$ with rounding scheme) of unit readings. We carried out 2 groups of simulations: one group (top row in Figure 10) used zeros to replace the "damaged" units; the other group (bottom row) made them output random values from the range of $[0, 1]$. For each malfunction ratio, five sets of "broken" units were picked, so that each data point is in-fact the average test accuracy of $10 \times 5 = 50$ simulations.

Figure 10 evidently validates that our 3T-RTCN provided much more superior fault-tolerance ability (to sensor malfunction) than other methods. Temporal threads might hit a better score at low malfunction rates, but the spatial thread surpasses temporal ones and possesses stronger discrimination power at a higher malfunction ratio. Another instructive and illuminating discovery is that all models (esp. JKSC and LDS-Martin) can better cope with the zero signal than the noise "generated" by malfunction units; it hence implies

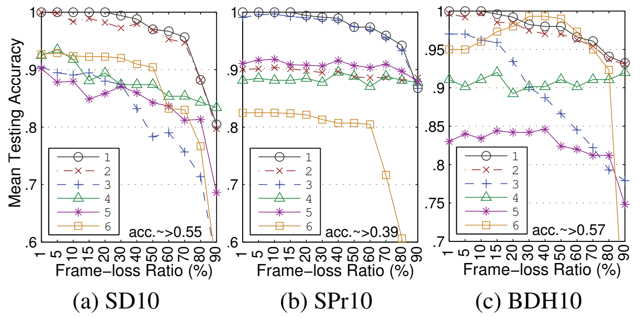


Figure 11: Fault-tolerance to frame loss: cf. legend in Fig. 9.

detecting malfunction units and forcing them to output zero values might be a beneficial approach for tactile recognition.

Fault-Tolerance to Frame Loss Another repeatedly seen corruption of tactile sensory data is *frame loss*, which is usually caused by *transmission circuit malfunction*. Frame loss occurs when some frame slices sent across the transmission circuit fail to reach their destinations. To simulate frame-loss context, we arbitrarily removed a ratio of frames from each sequence; and for every frame-loss ratio, we made 5 random choices of throwaway frames. From Figure 11, we found that spatial thread is more tolerable to frame-loss than temporal threads, because frame-loss primarily ruins the temporal coherence between neighboring frames. The performance of 3T-RTCN gently and monotonically decreased upon higher frame-loss ratio; nevertheless it performed constantly better than other models at almost any presumed frame-loss ratio. However, ST-HMP was hardly impacted by frame-loss; hence in this sense, it is a stable feature learning algorithm for tactile sequences. Interestingly, the frame-loss tolerance trend of JKSC varied dramatically on different datasets; its test accuracy could even conspicuously go up with more lost frames (e.g. BDH10); our guess is that frame-loss works like subsampling that eliminates the local peak and trough of tactile output over time axis (e.g. Fig. 4a); but it is hard to take this advantage, since over-subsampling can do great harm.

Conclusions and Perspectives

Inspired by the biological discovery of *segregated spatio-temporal neural pathways* for prefrontal control of fine tactile discrimination (Gogulski et al. 2013), we put forward the *segregated spatio-temporal RTCN threads* that constitute our 3T-RTCN model, which performs spatio-temporal feature representation and fusion for tactile recognition. It outperformed several state-of-the-art methods by a large margin on training efficiency, prediction accuracy, robustness, and fault-tolerance. In general, temporal threads tend to be more discriminative than spatial ones on sparse and filtered tactile signal. In comparison with convolutional weight-tuning and 3D ConvNet, our approach inevitably reduced training time dramatically, making it less challenging in converting 3T-RTCN to an equivalent online learning algorithm. We also noted that forcing the malfunction units generating zero values is potentially beneficial in enhancing the fault-tolerance ability. Our future perspectives include addressing larger-scaled tactile data, investigating the effectiveness of RNN (Recurrent Neural Networks), and a much more extensive analysis on computational complexity for both batch and on-line/sequential version of our model.

Acknowledgments

This work was supported by the grants from China National Natural Science Foundation under Grant No. 613278050 & 61210013. We express our appreciation to Guangbin Huang, Zuo Bai, and Andrew Saxe for the constructive suggestions. We thank Jingwei Yang and Rui Ma for the explanation of JKSC and BoS-LDSs source code. We also express our gratitude to Xiaohui Hu and Haolin Yang for their help in collecting the HCs10 dataset. Lele Cao is supported by the State Scholarship Fund under File No. 201406210275.

References

Anselmi, F.; Leibo, J. Z.; Rosasco, L.; Mutch, J.; Tacchetti, A.; and Poggio, T. 2015. Unsupervised learning of invariant representations. *Theoretical Computer Science* DOI:10.1016/j.tcs.2015.6.48.

Bekiroglu, Y.; Laaksonen, J.; Jorgensen, J. A.; Kyrki, V.; and Kragic, D. 2011. Assessing grasp stability based on learning and haptic data. *Robotics, IEEE Transactions on* 27(3):616–629.

Bekiroglu, Y.; Kragic, D.; and Kyrki, V. 2010. Learning grasp stability based on tactile data and HMMs. In *Proc. of the 19th Int'l Conf. on RO-MAN*, 132–137. Viareggio, Italy: IEEE.

Bicchi, A.; Scilingo, E. P.; Ricciardi, E.; and Pietrini, P. 2008. Tactile flow explains haptic counterparts of common visual illusions. *Brain research bulletin* 75(6):737–741.

Cao, L.; Kotagiri, R.; Sun, F.; Li, H.; Huang, W.; and Zay, M. 2015. Supplemental materials to this paper. http://escience.cn/peo-ple/caolele.

Chitta, S.; Piccoli, M.; and Sturm, J. 2010. Tactile object class and internal state recognition for mobile manipulation. In *Proc. of the 27th ICRA*, 2342–2348. Anchorage, Alaska: IEEE.

Drimus, A.; Kootstra, G.; Bilberg, A.; and Kragic, D. 2014. Design of a flexible tactile sensor for classification of rigid and deformable objects. *Robotics and Autonomous Systems* 62(1):3–15.

Gogulski, J.; Boldt, R.; Savolainen, P.; Guzmán-López, J.; Carlson, S.; and Pertovaara, A. 2013. A segregated neural pathway for prefrontal top-down control of tactile discrimination. *Cerebral Cortex (New York, NY: 1991)* 25(1):161–166.

Hinton, G. E. 2012. A practical guide to training restricted boltzmann machines. In *Neural Networks: Tricks of the Trade*. 599–619.

Horn, B. K., and Schunck, B. G. 1981. Determining optical flow. *Artificial Intelligence* 17:185–203.

Huang, G.-B.; Zhou, H.; Ding, X.; and Zhang, R. 2012. Extreme learning machine for regression and multiclass classification. *SMC, Part B: Cybernetics, IEEE Trans. on* 42(2):513–529.

Huang, G.-B.; Bai, Z.; Kasun, L. L. C.; and Vong, C. M. 2015. Local receptive fields based extreme learning machine. *IEEE Computational Intelligence Magazine* 10(2):18–29.

Jarrett, K.; Kavukcuoglu, K.; Ranzato, M.; and LeCun, Y. 2009. What is the best multi-stage architecture for object recognition? In *Proc. of the 12th CVPR*, 2146–2153. Miami, Florida: IEEE.

Ji, S.; Xu, W.; Yang, M.; and Yu, K. 2013. 3D convolutional neural networks for human action recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 35(1):221–231.

Johnson, B. W. 1984. Fault-tolerant microprocessor-based system. *IEEE Micro* 4(6):6–21.

Khaleghi, B.; Khamis, A.; Karray, F. O.; and Razavi, S. N. 2013. Multisensor data fusion: A review of the state-of-the-art. *Information Fusion* 14(1):28–44.

Kokar, M. M.; Tomasik, J. A.; and Weyman, J. 2004. Formalizing classes of info. fusion systems. *Information Fusion* 5(3):189–202.

Le, Q. V.; Zou, W. Y.; Yeung, S. Y.; and Ng, A. Y. 2011. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *Proc. of the 14th CVPR*, 3361–3368. Colorado Springs: IEEE.

Lee, H.; Grosse, R.; Ranganath, R.; and Ng, A. Y. 2009. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proc. of the 26th ICML*, 609–616. Montreal, Quebec: ACM.

Liu, H.; Greco, J.; Song, X.; Bimbo, J.; Seneviratne, L.; and Althoefer, K. 2012. Tactile image based contact shape recognition using neural network. In *Proc. of the Int'l Conf. on MFI*, 138–143. Hamburg, Germany: IEEE.

Ma, R.; Liu, H.; Sun, F.; Yang, Q.; and Gao, M. 2014. Linear dynamic system method for tactile object classification. *Science China Information Sciences* 57(12):1–11.

Madry, M.; Bo, L.; Kragic, D.; and Fox, D. 2014. ST-HMP: Unsupervised spatio-temporal feature learning for tactile data. In *Proc. of the 31st ICRA*, 2262–2269. Hong Kong, China: IEEE.

Ngiam, J.; Chen, Z.; Chia, D.; Koh, P. W.; Le, Q. V.; and Ng, A. Y. 2010. Tiled convolutional neural nets. *Adv. in NIPS* 1279–1287.

Pezzementi, Z.; Plaku, E.; Reyda, C.; and Hager, G. D. 2011. Tactile-object recognition from appearance information. *Robotics, IEEE Transactions on* 27(3):473–487.

Pezzementi, Z.; Reyda, C.; and Hager, G. D. 2011. Object mapping, recognition, and localization from tactile geometry. In *Proc. of the 28th ICRA*, 5942–5948. Shanghai, China: IEEE.

Pinto, N.; Doukhan, D.; DiCarlo, J. J.; and Cox, D. D. 2009. A high-throughput screening approach to discover good forms of biologically inspired visual representation. *PLoS computational biology* 5(11):e1000579(1–12).

Saisan, P.; Doretto, G.; Wu, Y. N.; and Soatto, S. 2001. Dynamic texture recognition. In *Proc. of the 4th CVPR*, volume 2, 58–63. Kauai, Hawaii: IEEE.

Saxe, A.; Koh, P. W.; Chen, Z.; Bhand, M.; Suresh, B.; and Ng, A. Y. 2011. On random weights and unsupervised feature learning. In *Proc. of the 28th ICML*, 1089–1096. Bellevue, WA: ACM.

Schmidt, M. 2012. minFunc: unconstrained differentiable multi-variate optimization in Matlab. Url http://www.di.ens.fr/mschmidt/ software/minfunc.html.

Schneider, A.; Sturm, J.; Stachniss, C.; Reisert, M.; Burkhardt, H.; and Burgard, W. 2009. Object identification with tactile sensors using bag-of-features. In *Proc. of the 21st Int'l Conf. on IROS*, 243–248. St. Louis, Missouri: IEEE/RSJ.

Simonyan, K., and Zisserman, A. 2014. Two-stream convolutional networks for action recognition in videos. *Adv. in NIPS* 568–576.

Soh, H.; Su, Y.; and Demiris, Y. 2012. Online spatio-temporal gaussian process experts with application to tactile classification. In *Proc. of the 24th IROS*, 4489–4496. Algarve, Portugal: IEEE.

Sun, D.; Roth, S.; and Black, M. J. 2010. Secrets of optical flow estimation and their principles. In *Proc. of the 23rd CVPR*, 2432–2439. San Francisco, California: IEEE.

Taylor, G. W.; Fergus, R.; LeCun, Y.; and Bregler, C. 2010. Convolutional learning of spatio-temporal features. In *Proc. of the 11th ECCV*, 140–153. Crete, Greece: Springer.

Tegin, J., and Wikander, J. 2005. Tactile sensing in intelligent robotic manipulation-a review. *Industrial Robot* 32(1):64–70.

Teshigawara, S.; Tadakuma, K.; Ming, A.; Ishikawa, M.; and Shimojo, M. 2010. High sensitivity initial slip sensor for dexterous grasp. In *Proc. of the 27th ICRA*, 4867–4872. Alaska: IEEE.

Xiao, W.; Sun, F.; Liu, H.; and He, C. 2014. Dexterous robotic hand grasp learning using piecewise linear dynamic systems model. *Foundations and Practical Applications of CSIP* 845–855.

Yang, H.; Liu, X.; Cao, L.; and Sun, F. 2015a. A new slip-detection method based on pairwise high frequency components of capacitive sensor signals. In *Proc. of the 5th ICIST*, 56–61. Changsha, China: IEEE.

Yang, J.; Liu, H.; Sun, F.; and Gao, M. 2015b. Tactile sequence classification using joint kernel sparse coding. In *Proc. of the 28th IJCNN*, 1–6. Killarney, Ireland: IEEE.